Ordering and Making Pizza for 120 Million

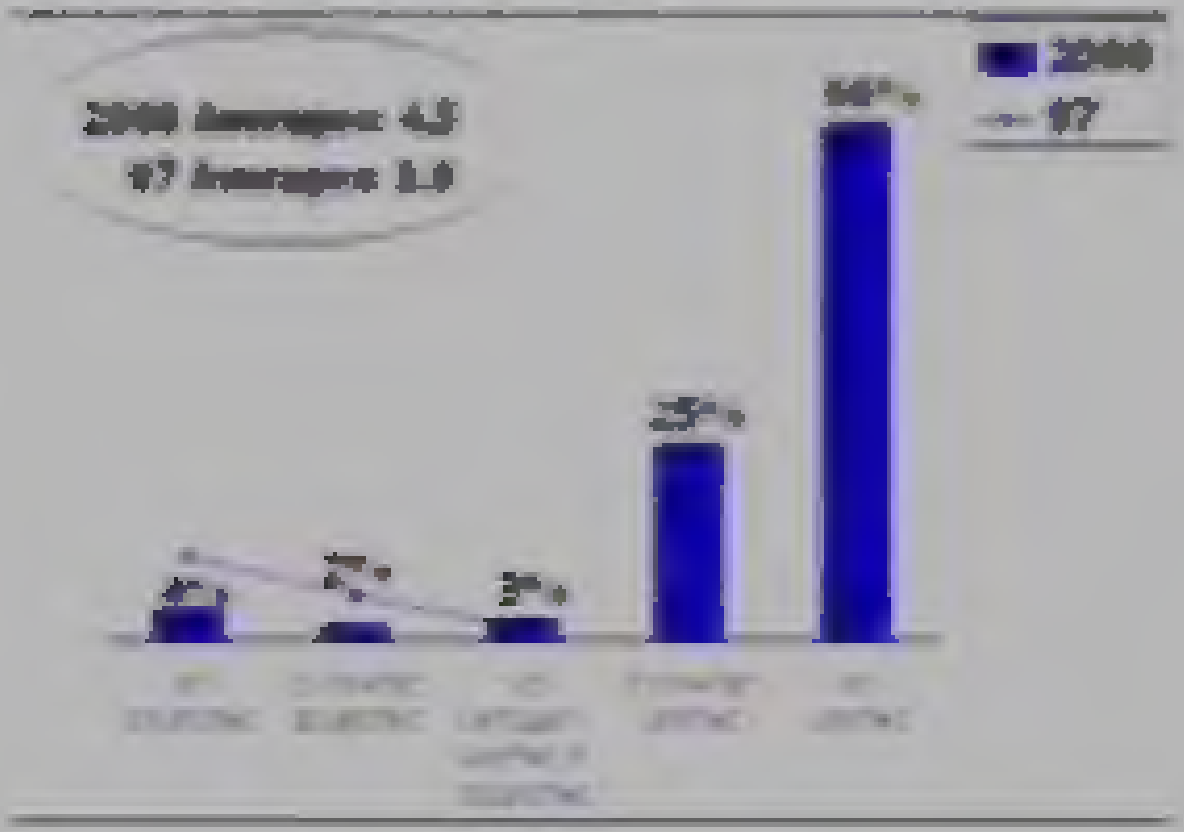
Designing and Building the Next Office

Agenda

- Designing a New Office
  - Office customer challenges
  - Understanding our customers
  - Using customer feedback, understanding in design
- Building Office
  - How we're organized
  - The development cycle
  - The code
  - The life of an Office developer
  - What we do well
  - What we don't do so well

Challenges – The Pizza Problem…

- 120 million relatively happy and loyal customers
  - Why should I get the next version?



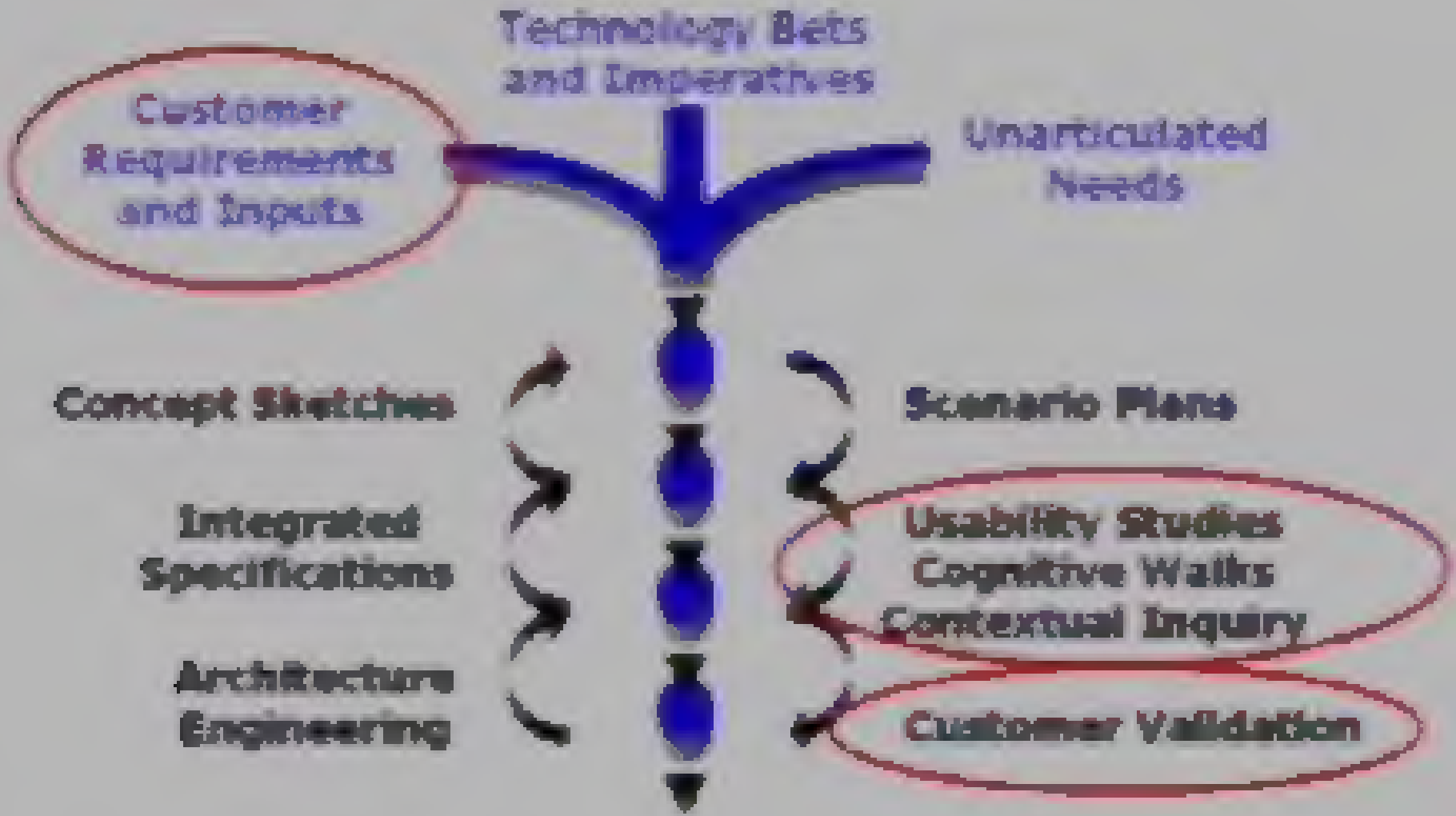
Challenges – The Pizza Problem...

- Wide spectrum of target customers
  - From end-users to administrators
- Want new features but no new learning
  - How do I migrate the users and the current learning
- Must be innovative but fully compatible
  - New features but no surprises (file formats, macros, solutions)
- Designed to be flexible but lockable
  - Fully customizable and configurable – but lockable if I so wish
- Fully loaded but not bloated
  - My 20% of the features are most important
  - The other 80% should not get in my way

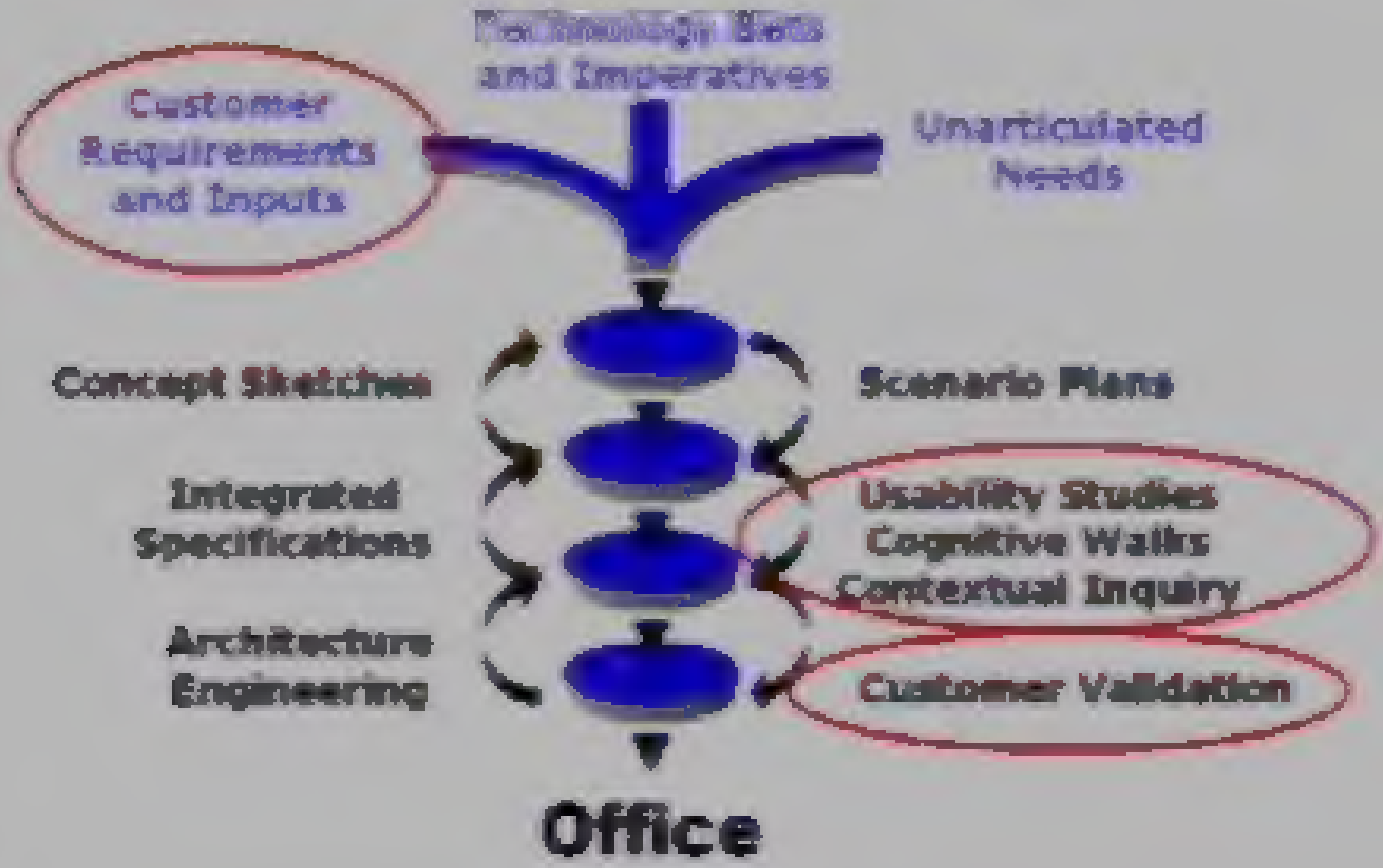
Understanding the customer better

- Instrumented versions
  - Log everything done by users
  - Some "duhs", some surprises
- Usability studies
  - Get to see and feel the customer's pain
- Customer visits
  - Everyone does them
  - Watch users work
- Advisory Council
  - From 15 hand-picked companies
  - Meet in Redmond before and during dev cycle

Designing for innovation



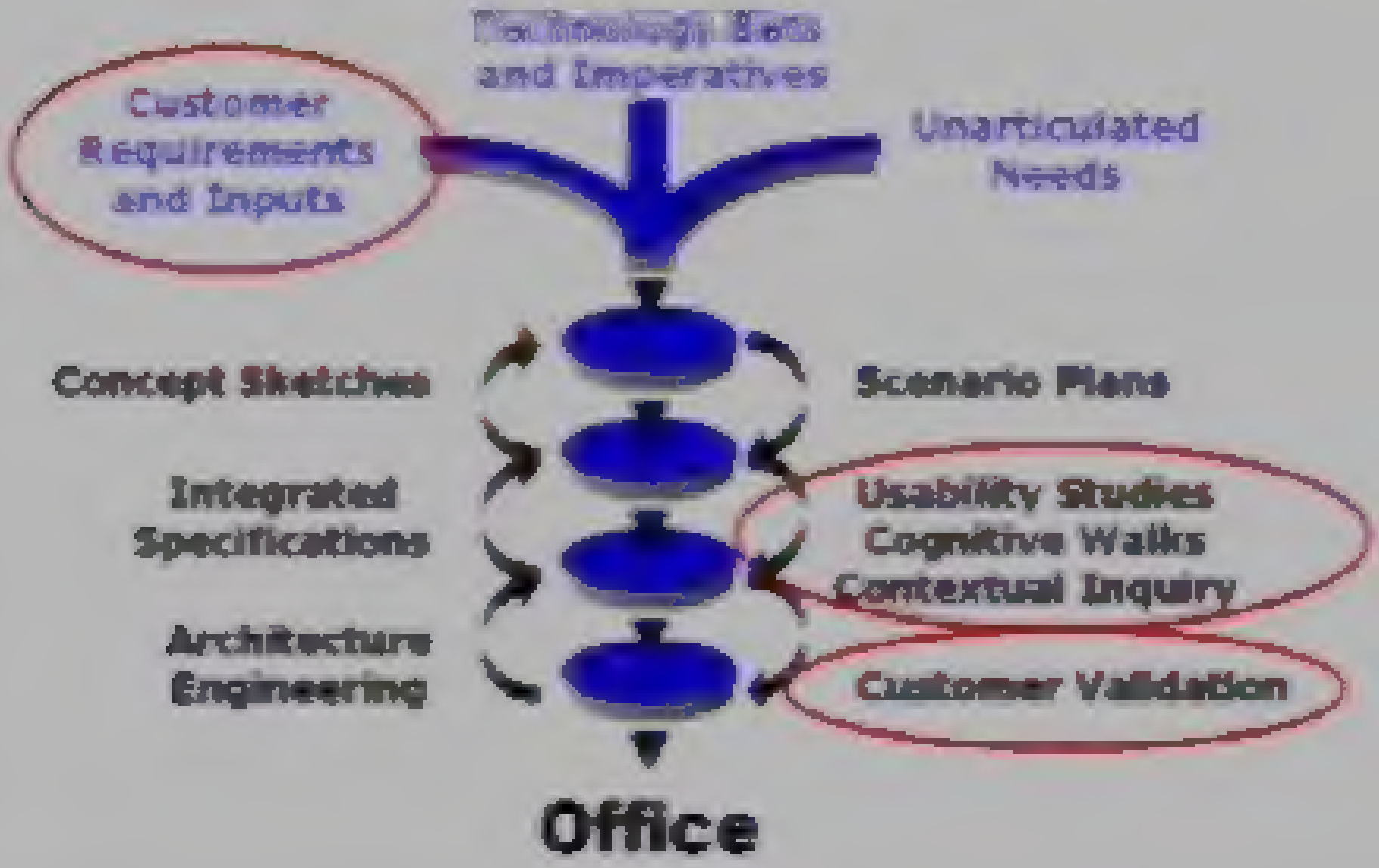
Designing for innovation



How We're Organized

- Roles:
  - Core:
    - Dev, Test, Prog. Mgmt., User Ed., Marketing
  - Others:
    - Product Planners
    - Designers
    - Usability Specialists
    - PSS
    - Localizers

Designing for innovation



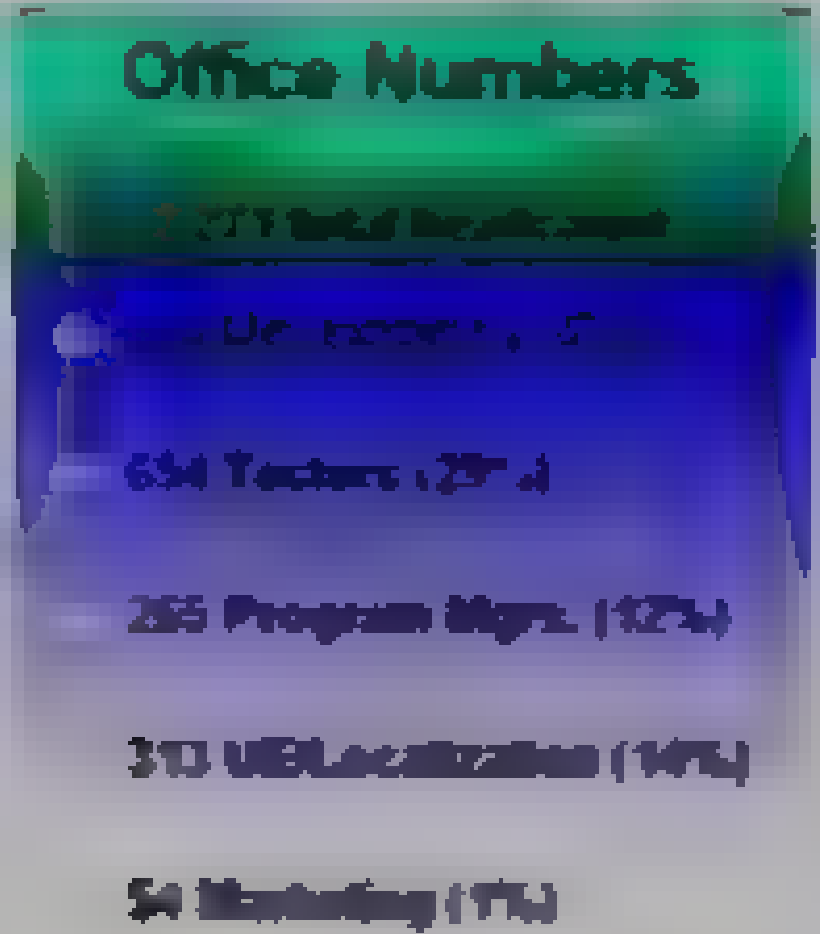
How We're Organized

- Roles:
  - Core:
    - Dev, Test, Prog. Mgmt., User Ed., Marketing
  - Others:
    - Product Planners
    - Designers
    - Usability Specialists
    - PSS
    - Localizers

How We're Organized

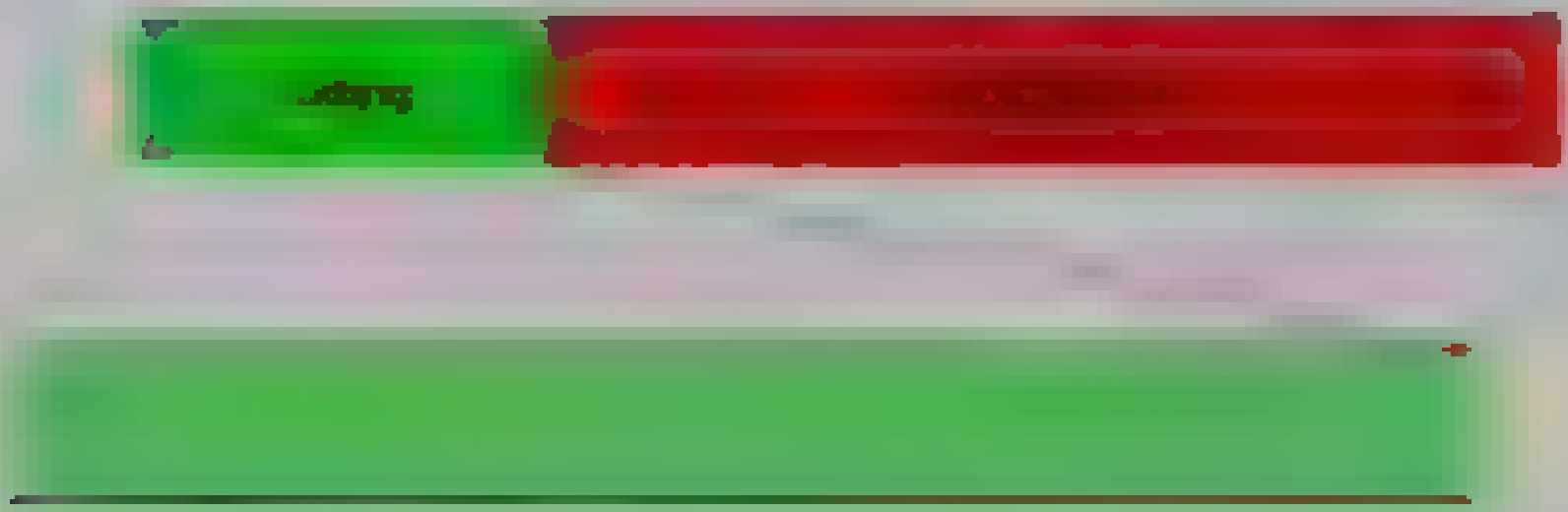
- Groups
  - [illegible]
    - [illegible]
    - [illegible]
  - [illegible]
    - [illegible]
    - [illegible]
    - [illegible]
  - [illegible]
    - [illegible]
  - [illegible]
    - [illegible]

How We're Organized

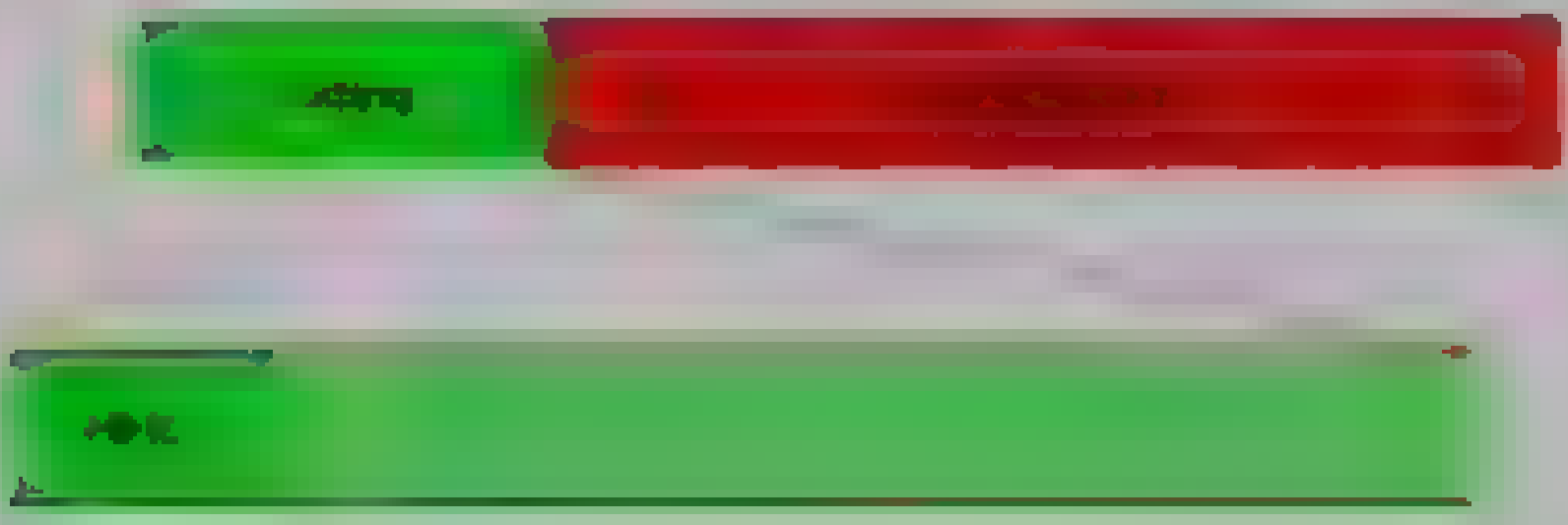


Development Timeline

Development Timeline

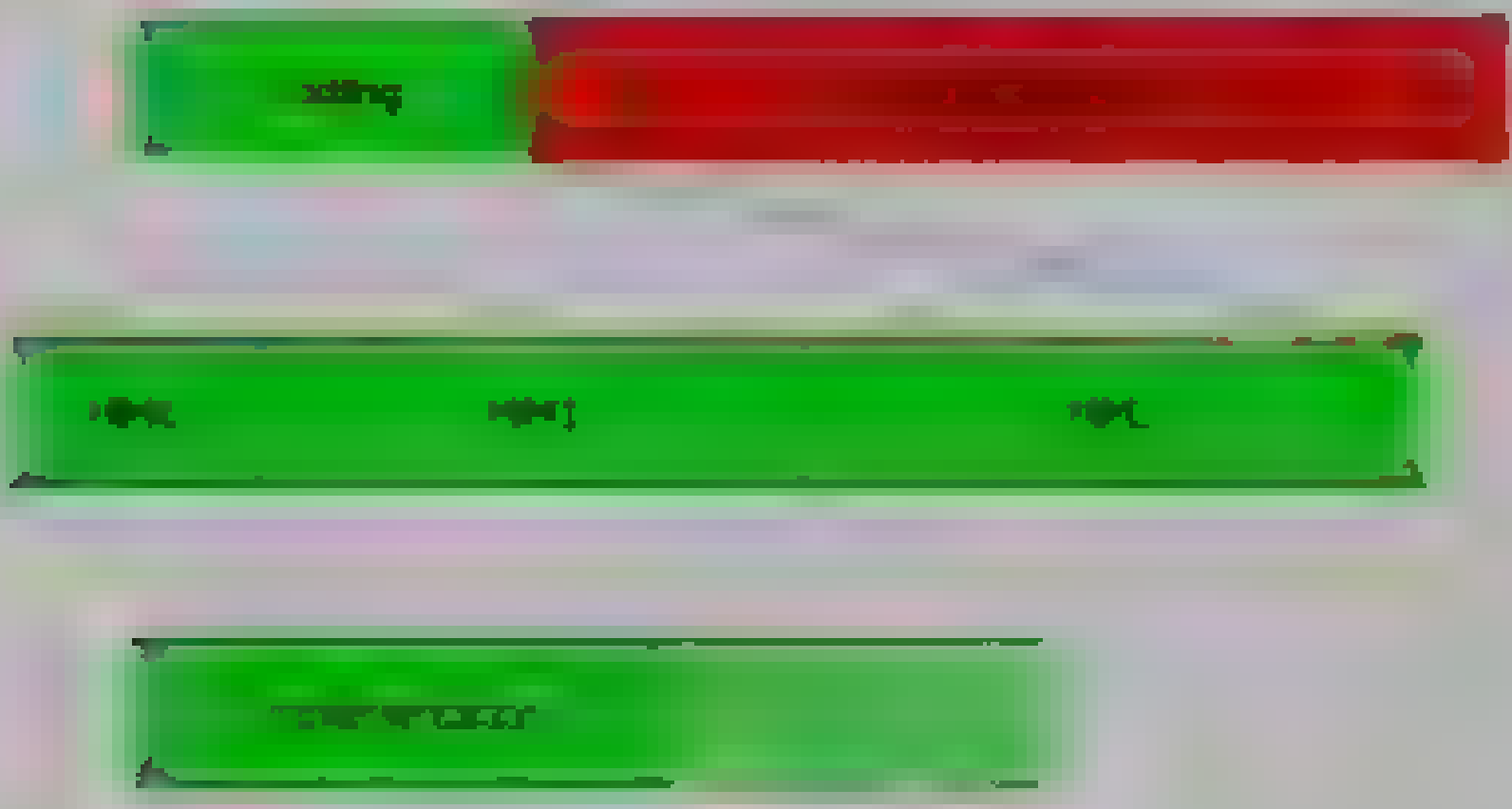


Development Timeline

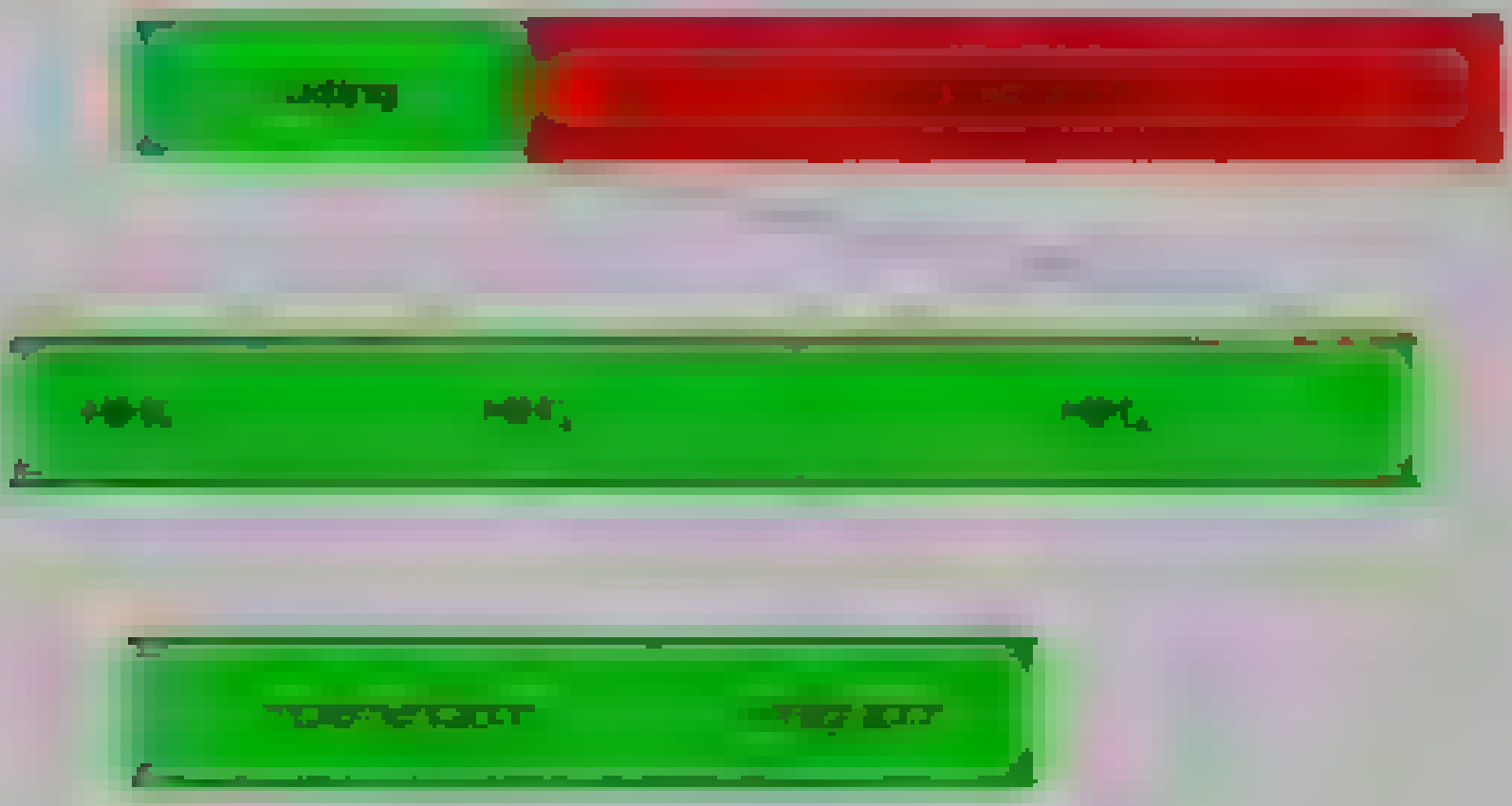


Development Timeline

Development Timeline



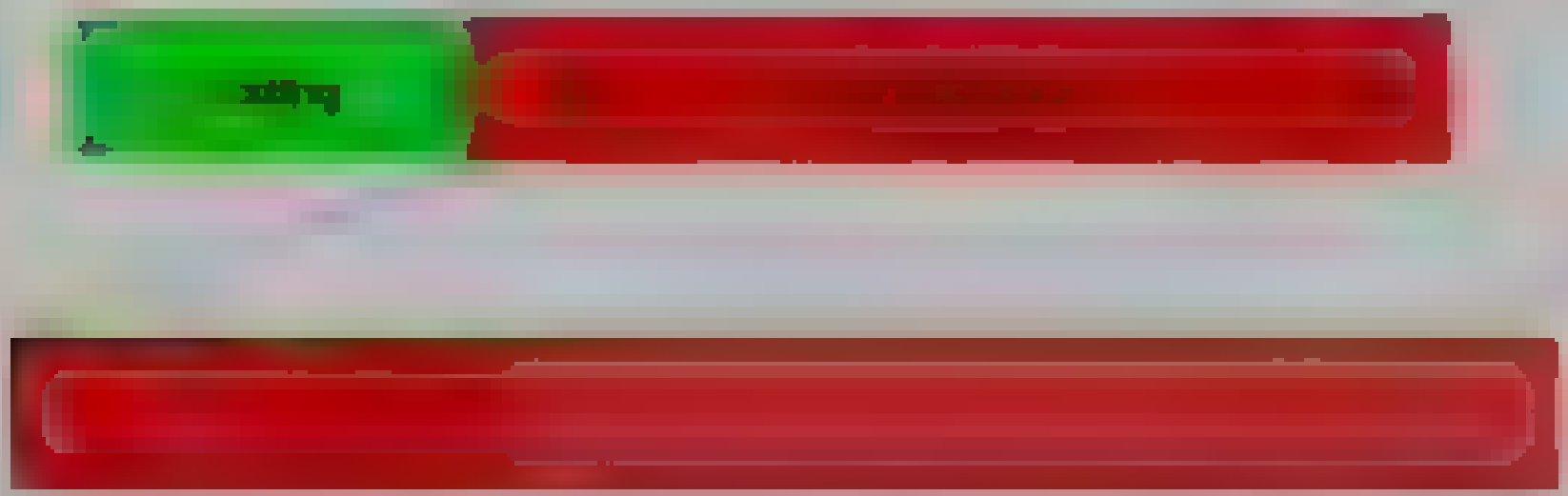
Development Timeline



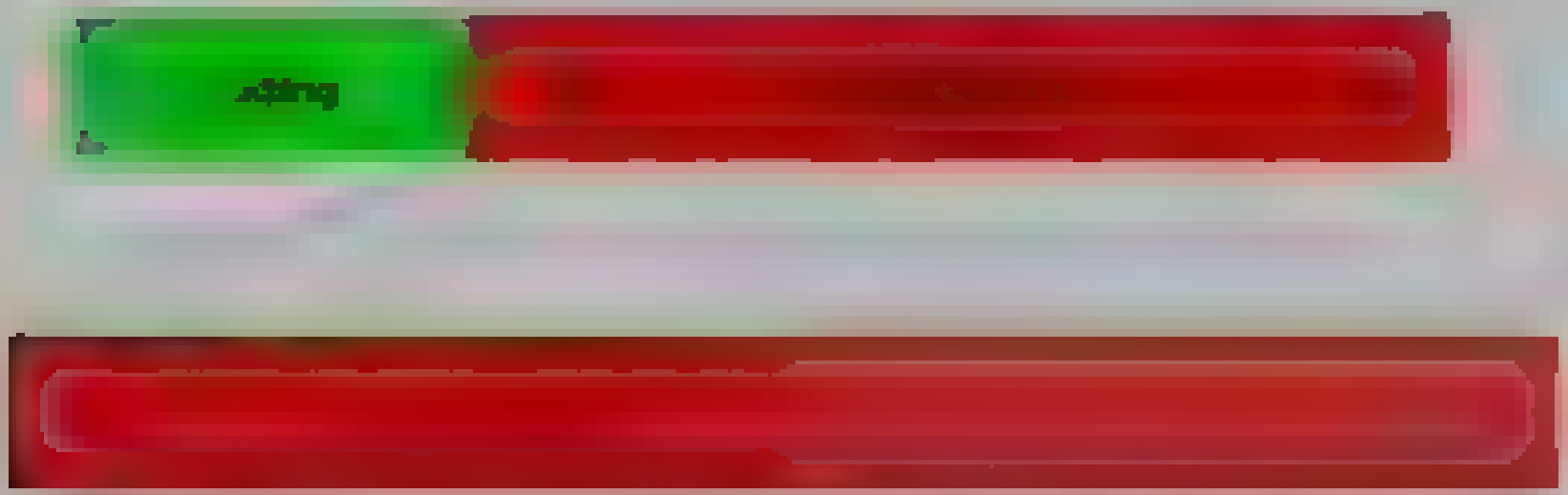
Development Timeline

Development Timeline

Development Timeline



Development Timeline



Development Timeline

Development Timeline

Development Timeline

Development Timeline

The life of an Office developer

- Features are assigned

The life of an Office developer

- Features are assigned
- Designs are done
- Design reviews
- Features are scheduled
- Schedule is rolled up and balanced

The life of an Office developer

The life of an Office developer

- [illegible]
- [illegible]
- [illegible]
  - [illegible]
    - [illegible]
  - [illegible]
    - [illegible]
    - [illegible]
    - [illegible]
  - [illegible]

The life of an Office developer

The life of an Office developer

- [illegible]
- [illegible]
  - [illegible]
  - [illegible]
  - [illegible]
- [illegible]
- [illegible]

About the Code

- All apps written in C
  - Excel, Word
- or in C++
  - MSC Access is C++ Code

About the Code

How much code is there?

About the Code

How much code is there?

Office 2000	
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
Total	**7,705,138**

About the Code

How much code is there?

Office 2000	
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
Total	**7 705 138**

Office 97	
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
Total	**4 056 270**

About the Code

- Word in Office 2000
  - [illegible]
  - [illegible]
  - [illegible]
  - [illegible]
  - [illegible]

What we do well

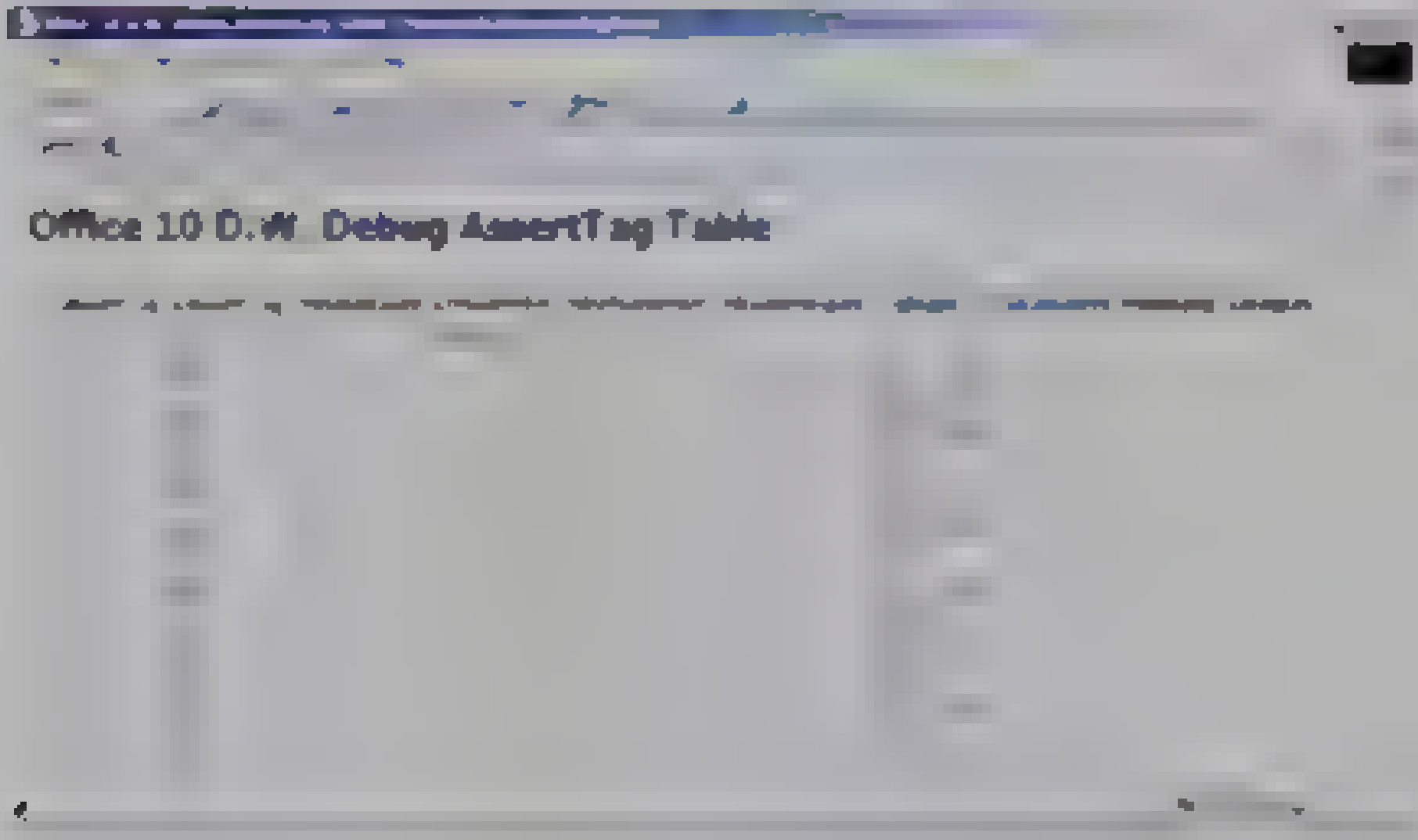
- [illegible]
- [illegible]
- [illegible]
  - [illegible]
- [illegible]
  - [illegible]
- [illegible]
- [illegible]
  - [illegible]
  - [illegible]
  - [illegible]

http://office10

What we do well

- [illegible]
  - [illegible]
  - [illegible]
- [illegible]
  - [illegible]
  - [illegible]
- [illegible]
  - [illegible]
  - [illegible]

AssertTags



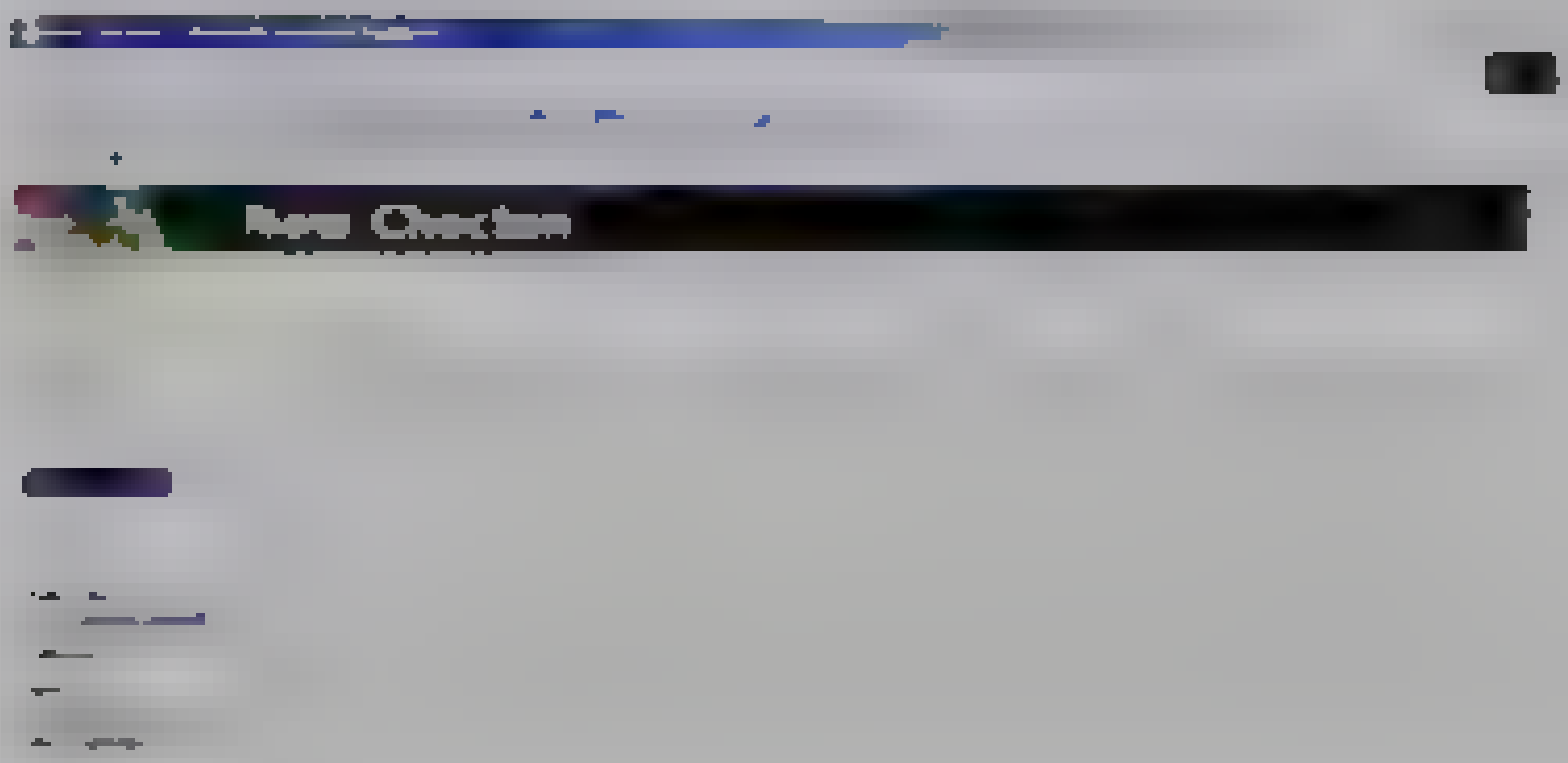
AssertTags (cont.)

OfficeWatson

What we do well

- Easy to use performance tools
  - [illegible]
  - [illegible]
- Coding Scheduling methodologies
  - [illegible]
  - [illegible]
  - [illegible]
    - [illegible]
    - [illegible]
    - [illegible]

Check-in tool



Things we're not good at

- Maximizing coding time
  - Coding to overhead ratio too high
  - Need more robust builds/tests
    - Too much time blocked
    - Too many "random" failures today
    - Some tools are too fragile
    - The "environment" is too fragile
  - Build break resolution/turn-around timeliness

Things we're not good at

- How do we reduce bug fixing time?
  - Improve checkin code quality
    - Verification tools
    - Better developer discipline
    - Pre-checkin test depth/targeting
    - Staging and sub-projects
  - Reduce the number of "spec issue" bugs
    - One analysis shows this is 40% of all bugs

Things we're not good at

- Reducing bug fixing time (cont.)
  - Need better way of prioritizing bugs
    - Gather more data internally & from betas
  - Performance roller coaster
    - Too much "break it now, fix it later"

Things we're not good at

- Schedule predictability
  - Better data
  - Better model of how things really are
  - Schedule *everything*